SRI International

OVERVIEW OF THE IMAGE UNDERSTANDING TESTBED

Technical Note 310

October 1983

By: Andy Hanson
        Sr. Computer Scientist
        Artificial Intelligence Center
        Computer Science and Technology Division

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

Report Documentation Page	*Form Approved* *OMB No. 0704-0188*

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **OCT 1983**	2. REPORT TYPE	3. DATES COVERED **00-10-1983 to 00-10-1983**

4. TITLE AND SUBTITLE **Overview of the Image Understanding Testbed**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **SRI International,333 Ravenswood Avenue,Menlo Park,CA,94025**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**25**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

I   INTRODUCTION

The Image Understanding Testbed is a system of hardware and software that is designed to facilitate the integration, testing, and evaluation of implemented research concepts in machine vision. The system was developed by the Artificial Intelligence Center of SRI International under the joint sponsorship of the Defense Advanced Research Projects Agency (DARPA) and the Defense Mapping Agency (DMA). The primary purpose of the Image Understanding (IU) Testbed is to provide a means for transferring technology from the DARPA-sponsored IU research program to DMA and other organizations in the defense community.

The approach taken to achieve this purpose has two components:

* The establishment of a uniform environment that will be as compatible as possible with the environments of research centers at universities participating in the IU program. Thus, organizations obtaining copies of the testbed can receive new results of ongoing research as they become available.

* The acquisition, integration, testing, and evaluation of selected scene analysis techniques that represent mature examples of generic areas of research activity. These contributions from IU program participants will allow organizations with testbed copies to immediately begin investigating potential applications of IU technology to problems in automated cartography and other areas of scene analysis.

An important component of the DARPA IU research program is the development of image-understanding techniques that could be applied to automated cartography and military image interpretation tasks; this work forms the principal focus of the testbed project. A number of computer modules developed by participants in the Image Understanding program have been transported to the uniform testbed environment as a first step in the technology transfer process. These include systems written in UNIX C, MAINSAIL, and FRANZ LISP. Capabilities of the computer programs include segmentation, linear feature delineation, shape detection, stereo reconstruction, and rule-based recognition of classes of three-dimensional objects.

1

A   Documentation

The following documents relating to the IU testbed are now available as SRI technical notes:

* "The DARPA/DMA Image Understanding Testbed User's Manual" presents a user's view of the testbed. It outlines the general structure of the system and describes the use of major facilities.

* "The IU Testbed Programmer's Manual" collects UNIX-style "man" pages describing testbed programs, libraries, and files.

* "The DARPA/DMA Image Understanding Testbed System Manager's Manual" contains information relevant to system implementation and management issues.

* "Managing the IU Testbed under EUNICE/VMS" provides specific details for managers and users of systems that run the EUNICE/VMS emulation of the UNIX operating system.

The following reports evaluating major contributed software systems are also available:

* GHOUGH: "The GHOUGH Generalized Hough Transform Package: Description And Evaluation"

* PHOENIX: "The PHOENIX Image Segmentation Package: Description And Evaluation"

* RELAX: "The RELAX Image Relaxation System: Description And Evaluation"

Documentation describing the CMU-contributed graphics and picture-file access systems is provided in the following separate CMU documents:

* CMU002: "Grinnell Display Software Support"

* CMU003: "CMU Image Format and Paging System"

* CMU004: "Image File Naming Conventions."

B.   Hardware Configuration

The principal elements of the IU testbed hardware configuration are a DEC VAX-11/780 central processing unit, with its peripherals, and several Symbolics Model 3600 Lisp Machines. The SRI testbed VAX is a four-megabyte system with one tape drive, four 300-MB disk drives, one 414-MB Winchester drive, and 32 teletype lines. The VAX interfaces directly to a variety of terminals. Graphics capabilities are provided by Grinnell and DeAnza display systems, both with 512 x 512 resolution and full color support. Several kinds of pointing devices, such as "mice" and digitizing tablets, are available. Other peripherals include a Versatec 11-inch printer/plotter with 200-point/inch resolution (which functions as a phototypesetter) and an Optronics C-4100 color image scanner with resolution selectable from 12.5 to 400 microns. The testbed system also supports an ARPANET network link with network address SRI-IU.

Each Lisp Machine has 2-MB of memory and a 180-MB disk drive. A 10-Mbit/second Ethernet network connects the Lisp Machines to one another and to the VAX. Color graphics systems and additional disk drives may eventually be added to enhance the capabilities of the testbed Lisp Machine environment.

C.   Operating-System Software

The Image Understanding Testbed system may be run under either the UNIX* operating system or under the VAX/VMS** operating system. In principle, all testbed applications software can be run on either UNIX or VMS/EUNICE*** systems, provided that appropriate system-specific hardware device drivers are available.

A "32V," or higher, UNIX license is required to operate the testbed under either system; in addition, a EUNICE license is needed to run the testbed under VAX/VMS.

* UNIX is a trademark of Bell Laboratories.

** VAX/VMS is a trademark of the Digital Equipment Corporation.

*** EUNICE is a proprietary software product of SRI International.

The testbed currently uses the Berkeley VAX/UNIX 4.1c BSD system software distribution with support for the IP/TCP networking protocols; 4.2 BSD will be supported when it becomes generally available. UNIX device drivers are supplied by Berkeley for the Versatec printer/plotter and for ARPANET devices; a UNIX driver for the Grinnell display system has been provided by CMU. No Optronics scanner driver is available under UNIX at this time.

Under the VAX/VMS operating system, UNIX is emulated by the EUNICE system. This combination of operating-system support permits compatibility with both UNIX and other VMS/EUNICE environments. VMS device drivers are currently available for ARPANET devices, the Grinnell display system, the Versatec printer/plotter, and the Optronics image scanner. See the document "Managing the IU Testbed under EUNICE/VMS" for further details.

D.   Languages

The principal high-level programming languages on the testbed VAX are UNIX C and FRANZ LISP. MAINSAIL, an ALGOL-like language, is available under both UNIX and VMS, but is currently used only on the SRI EUNICE/VMS testbed system. Other LISP dialects that may be used on the testbed include ISI VAX INTERLISP and MIT NIL VAX LISP. FORTRAN and PASCAL compilers are available under both UNIX and VMS, but are not used in any contributed software. On EUNICE/VMS systems, the DEC C-language compiler can be used instead of the UNIX C compiler for some applications; although the DEC C compiler generates exceptionally efficient code, substantial changes may be required to compile and run code written originally for a UNIX C system.

Graphics functions on the testbed Grinnell display are fully supported in C; the testbed software is based on the CMU Grinnell graphics package. Supplementary testbed graphics capabilities are available for the DeAnza in MAINSAIL. FRANZ LISP and MAINSAIL programs may access the Grinnell by means of the C-language Grinnell graphics package.

4

Lisp Machine LISP is of course available on the Lisp Machines, which have now been integrated into the testbed system.

II   CONTRIBUTED SOFTWARE

A.   Overview of Applications Software Contributed to the Testbed

Besides SRI International, the institutions contributing software systems to the DARPA/DMA Image Understanding Testbed are Carnegie-Mellon University (CMU), the Massachusetts Institute of Technology (MIT), Stanford University, the University of Maryland, the University of Rochester, and the University of Southern California (USC). Modified or reimplemented versions of some routines have also been provided by a DARPA project at Hughes Aircraft Corporation.

Software modules integrated into the testbed include main programs, program systems, libraries of user utilities, graphics routines, and image access routines. Each of the designated testbed contributor sites has defined and delivered contributions to the testbed system. Among the research contributions are four modules from SRI and two from CMU; also running on the testbed are one contribution each from Rochester, Maryland, and USC, as well as a major system in FRANZ LISP from Stanford. MIT has provided a system in Lisp Machine LISP that runs on the testbed Lisp Machines. CMU has also furnished utilities, graphics, and picture access packages, while SRI has implemented an extended picture format and many additional utilities.

A summary of the currently operational research software contributions is given in Table I.

Table 1

SUMMARY OF TESTBED RESEARCH CONTRIBUTIONS

INSTITUTION	CONTRIBUTION	LANGUAGE
CMU	Picture access and display packages	C
	PHOENIX segmentation system	C
	Stereo/correlation system	C
MARYLAND	Relaxation package	C
MIT	Stereo reconstruction system	LISP MACHINE LISP
ROCHESTER	Generalized Hough transform system	C
SRI	Road expert	MAINSAIL
	RANSAC	MAINSAIL
	CAMDIST camera modeler	C
	SHOWDTM terrain map utility	C
STANFORD	ACRONYM 3-D model-based vision system	FRANZ LISP
USC	Linear-feature analysis	C

The following subsections summarize the status of each of the currently integrated contributions.

1. Carnegie-Mellon University Contributions

(1) CMU Grinnell Graphics and Image Manipulation Packages

* Date received: August 1981.
* Responsible party: David McKeown.
* Language: C (Berkeley UNIX) running on the VAX.
* Documentation: For complete documentation, see
   - CMU002: "Grinnell Display Software Support,"

6

- CMU003: "CMU Image Format and Paging System," and

- CMU004: "Image File Naming Conventions."

UNIX "man" entries providing high-level descriptions are available under the topics "cmuimglib," "gmrfrmlib," and "gmrlib." For testbed-based extensions to the CMU capabilities, see "dsplib," "frmlib," "imgfrmlib," "imglib," "imgnmelib," "piciolib," and "piclib."

* Description: These packages provide basic access to the functions of the Grinnell display system, as well as the capability of accessing image data files independently of the display system.

* Remarks: A number of minor modifications were needed to make the CMU package work with the SRI Grinnell configuration. The present code will support any CMU configuration or the SRI testbed configuration. The CMU image access package has also been integrated into the testbed environment; a new, extended testbed picture format has been implemented. Finally, there are several other general utilities of various sorts which have been supplied by CMU; see Section III.A.

(2) PHOENIX Segmentation Package

* Date received: December 1981.

* Responsible party: Steve Shafer.

* Language: C (Berkeley UNIX) running on the VAX.

* Usage: Invoke the command

        phoenix inimage -o outimage -f feat1 [feat2 ...]
                [-i file | -I file]
                [-e] [-s] [-O file -r reg -R reg]

* Documentation: A "man" entry for PHOENIX is available under the topic "phoenix." Testbed documentation is provided in "The PHOENIX Image Segmentation System: Description And Evaluation." See also "Recursive Region Segmentation by Analysis of Histograms," a CMU preprint by S. Shafer and T. Kanade.

* Description: PHOENIX performs image segmentation by recursive region splitting. This segmentation package uses the Ohlander histogram-partitioning method to segment color imagery. Each pixel in the input image is assigned a segment identification label according to the image characteristics and the parameters selected. Segmentation is carried out hierarchically, with higher-level regions segmented into subregions. Segmentation ceases in a given

7

region when the program criteria for significance of the next level of segmentation have not been met.

* Remarks: This system has a sophisticated user interface and a checkpoint mechanism.

(3) Stereo Reconstruction and Correlation Package

* Date received: September 1981.

* Responsible party: Charles Thorpe.

* Language: C (Berkeley UNIX) running on the VAX.

* Usage: Invoke either of the two commands

    correlate [-nqros v m t i ffilename -ffilename]

    stereo

and answer the prompts for additional program input parameters.

* Documentation: "man" entries for CORRELATE and STEREO are available under the topics "correlate" and "stereo." See also "Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover," Ph.D. Thesis by Hans Moravec.

* Description: This is a C version of the Moravec correlation and stereo reconstruction package written originally in SAIL at Stanford. The package consists of two portions: CORRELATE selects a set of "interesting" points in one image, using the Moravec interest operator, then attempts to locate the corresponding points in a second image by using an efficient hierarchical correlation matcher; STEREO uses the same method as CORRELATE to find corresponding points in a series of up to 9 images, then employs the Moravec method to assign a stereo depth value and confidence level to each match point.

* Remarks: This package implements all the basic capabilities of the original Moravec SAIL system, plus a number of enhancements introduced by Charles Thorpe.

2.   University of Maryland Contributions

Relaxation Package

* Date received: Final version received 9 July 1981.

* Responsible party: Bob Kirby (author: Russell Smith, revised by Joe Pallas).

8

*   Language: C (Berkeley UNIX) running on the VAX.

*   Usage: Invoke the command

    relax

    or invoke various elements of the package individually. The individual programs making up the system include:

    defcom defnbr imgprb prbimg relax relaxpar setup.csh

*   Documentation: A "man" entry for RELAX is available under the topic "relax." Testbed documentation is provided in "The RELAX Image Relaxation System: Description and Evaluation."

*   Description: This relaxation package takes an initial set of probabilities that a pixel belongs to each of a set of classes and iteratively adjusts them according to the probabilities of neighboring pixels. Two options are provided: an additive Hummel-Zucker-Rosenfeld relaxation algorithm and a multiplicative Peleg relaxation algorithm. A utility is provided for generating a two-class set of probabilities based on the luminance values of an image; the inverse operation is available to generate a grey-scale image from the reassigned probabilities, so that the user may monitor the relaxation process visually.

*   Remarks: A multiclass method of generating probability assignments corresponding to luminance values has been added for test and demonstration purposes.

3. MIT Contributions

Marr-Poggio-Grimson Stereo System

*   Received: February 1983

*   Responsible parties: Mike Brady, Eric Grimson, and Keith Nishihara.

*   Language: Lisp Machine LISP.

*   Documentation: Current documentation consists of comments in the programs themselves. Additional documentation is planned by MIT.

*   Description: This system uses zero-crossing matches at several scales to compute disparity values between stereo pairs. Additional consistency checking is available as an option.

*   Remarks: This system makes use of an extensive package of Lisp Machine vision utilities, some generated at MIT and some revised or newly developed at SRI. In particular,

routines for reading and writing 8-bit images in testbed
format have been provided; images may be read and written
on the local Lisp Machine file systems, or may be read and
written across the local network to the testbed VAX. The
testbed Lisp Machine utility systems have been modified for
use with the Symbolics 3600 Lisp Machines and now run in
that environment. Convolutions are currently done in
software. To enhance performance, it would be desirable to
have convolution hardware on the Lisp Machines.

4. University of Rochester Contributions

Hough Transform Package

*   Date received: May 1981.

*   Responsible parties: Dana Ballard and Bill Lampeter.

*   Language: C (Berkeley UNIX) running on the VAX.

*   Usage: Invoke the command

                    ghough

    and answer the prompts for program input parameters.

*   Documentation: A "man" entry for GHOUGH is available under
    the topic "ghough." Testbed documentation is available in
    "The GHOUGH Generalized Hough Transform Package:
    Description And Evaluation."

*   Description: This program takes a geometric-shape template
    and attempts to find matching shapes in the image, using
    the generalized Hough transform technique. The matched
    shapes may differ in displacement, rotation, and scale from
    the supplied template. The most likely values of location,
    rotation angle, and scale are printed out and the
    reoriented templates are displayed over the image.

*   Remarks: The CMU graphics package has been used as a basis
    for incorporating full interactive graphics into this
    system for both template generation and picture processing.
    Several improvements have been made in the user interface
    as well as in the efficiency of the code. The package was
    extended to handle multiple instances of an object.

5. SRI Contributions

(1) Road Expert

10

* Date received: January 1981.

* Responsible parties: Lynn Quam and Helen Wolf

* Language: MAINSAIL running under EUNICE on the VAX.

* Usage: While connected to the /iu/sri/road/cmd directory, start up the MAINSAIL system and invoke the TRKACQ module.

* Documentation: A "man" page is available under the topic "road," along with demonstration instructions in the user's manual.

* Description: This package acquires and tracks linear features, such as roads, in aerial imagery. Tracking is done automatically in imagery with a known ground truth data base. Once a road has been identified and tracked, a separate subsystem is available to analyze road surface anomalies and to assign them to such categories as vehicles, road surface markings, and shadows.

(2) RANSAC Image-to-Data-Base Correspondence Package

* Date received: January 1981.

* Responsible parties: Martin Fischler and Robert Bolles.

* Language: MAINSAIL running under EUNICE on the VAX.

* Usage: While connected to the /iu/vision directory, start up the MAINSAIL system and invoke the INTMOD module.

* Documentation: A "man" page is available under the topic "ransac," along with demonstration instructions in the user's manual.

* Description: This package selects a best fit to an array of control points that may possibly contain gross errors. If such errors are present, RANSAC offers significant improvements over least-squares fitting techniques. A typical application is to compute the camera model from a given set of landmarks in aerial imagery.

(3) CAMDIST Camera Model System

* Date received: March 1983

* Responsible party: Marsha Jo Hannah

* Language: C (Berkeley UNIX) running on the VAX

* Usage: Invoke the command

                    camdist [options]

11

with desired options.

*   Documentation: A "man" entry is available under the topic "camdist."

*   Description: CAMDIST provides a facility for performing a generalized least-squares solution for the relative position and orientation angles between two cameras, given a series of points in the two camera views, and/or for using this information to calculate the distances to these points. Wild points are automatically edited out. Errors are propagated from the image plane points through the camera model to derive errors for the assigned distances.

(4) SHOWDTM Terrain Model System

*   Date received: February 1983

*   Responsible party: Marsha Jo Hannah

*   Language: C (Berkeley UNIX) running on the VAX

*   Usage: Invoke the command

                      showdtm [options]

with desired options.

*   Documentation: A "man" entry is available under the topic "showdtm."

*   Description: This is an interactive program for displaying a i ital terrain model and producing either a perspective grid plot or a perspective range image of a portion of a model. When invoked with no arguments, SHOWDTM will prompt for the name of a terrain model (an image in testbed format), then wait for commands. If the name of an image fi e is spe ifie , the program will open that file, then wait for commands. If an initial command string is specified, the program will execute each of those commands, then wait for more.

6.   Stanford University Contributions

ACRONYM System

*   Date received: March 1982.

*   Responsible parties: Tom Binford and Rod Brooks.

*   Language: FRANZ LISP running on the VAX.   An extensive macro package is used to preserve most of the original MACLISP code.

12

* Usage: While connected to the directory /iu/acronym/sys,
  invoke "acronym". Connect to the models directory using
  (chdir ´../models), invoke (PARSE model-file-name), and
  proceed with the desired ACRONYM process.

* Documentation: Some basic instructions are contained in
  /iu/acronym/info and are accessible by invoking the command
  file info.com; this command starts up an EMACS INFO system
  with a special ACRONYM node. Other information is
  available in /iu/acronym/doc. A "man" entry for ACRONYM is
  available under the topic "acronym." See also "ACRONYM:
  The Facts," a partially completed Stanford University
  document by Rodney Brooks. A more complete set of
  documentation will eventually be supplied by the Hughes
  Aircraft ACRONYM-based vision project.

* Description: ACRONYM takes a scene that has been reduced to
  a set of two-dimensional ribbons and searches for instances
  of three-dimensional models that have been supplied to the
  system as data. This is a rule-based system that allows
  great flexibility in interpretation and scene-prediction.
  Models can also be defined in a very general manner by
  using generalized cones, constraints, and subclass
  definitions.

* Remarks: Reduction of an image to a list of ribbons must
  now be done by hand, starting with a corresponding file of
  line segments generated by a program such as the Nevatia-
  Babu line finder. While some test imagery is available
  with the ribbon reduction already carried out, the testbed
  ACRONYM system would profit from the addition of an
  automated ribbon-reduction module. Such a module has been
  promised by the Hughes Aircraft project.

7.   University of Southern California Contributions

Nevatia-Babu Line Finder

* Date received: June 1981 (SAIL version); June 1982 (C
  version from Hughes Aircraft).

* Responsible parties: Ram Nevatia at USC; Julius Bogdanovich
  at Hughes Aircraft.

* Language: C (Berkeley UNIX) running on the VAX.

* Usage: Connect to the /iu/usc/tst directory and run the
  following programs in sequence:

                    ../bin/convolve
                    ../bin/thrin
                    ../bin/psmaker
                    ../bin/linkseg

13

The output on seg.dat may be put into a device-independent
display format by invoking ../bin/segdisp; the testbed
graphics utility can be used to show the resulting
display file.

*

VERSION OF THE NEVATIA-BABU LINEFINDER." A brief "man"
entry is available under the topic "line."

* Description: This package extracts linear features from an
image and produces a data base of line segments. The
testbed C version supports 5 x 5 convolution masks
configured to identify edges oriented at 30-degree
intervals. The edges are then linked together into chains
and broken into straight-line segments.

* Remarks: The C version of this package lacks the parallel-
line (APAR) and supersegment (SAP) extraction routines
present in the SAIL version. It would be useful for
purposes of comparison to have these capabilities
available. Support for using a variety of convolution
masks would also be desirable.

B.   Demonstration, Test, and Evaluation of Testbed Modules

The final stage of the testbed project at SRI was the
demonstration, testing, and evaluation of the contributed software
modules. Our purpose here was twofold:

* To provide information that would be useful for assessing
the relevance of software techniques represented in the
testbed.

* To establish a model for evaluation of comparable IU
software.

Each module supports a standard demonstration of its capabilities.
The degree to which testing and evaluation can be carried out
meaningfully depends on the flexibility of each individual program.
Some can run on completely arbitrary images, while others require
extensive supporting data that cannot be easily assembled for arbitrary
images. Furthermore, some contributions have been extensively
documented in existing literature, while others have required additional
modifications and documentation regarding their operation in the testbed
environment. Accordingly, we have divided the contributions into the
following two general classes:

14

(1) DEMONSTRATION ONLY. Several major stand-alone systems need customized data bases to function correctly. When tools for construction of such data bases are not available, the modules will run only on a limited set of images, thus restricting the nature of the evaluation that can be carried out. Such systems are not appropriate for systematic evaluation over large numbers of images because of the operational difficulties of setting up the required contexts. These systems are available for demonstration on limited data sets. Documentation of these systems on the testbed includes at least a "man" page and demonstration instructions in the user's manual. Some have additional manuals or are discussed extensively in the literature. Implementation details are generally undocumented, so users concerned with implementation methods must examine the software directly.

(2) DETAILED EVALUATION. Several modules that can readily be exercised on a wide variety of imagery have been subjected to rigorous investigation. Thorough evaluation reports have been prepared that describe the parameters, performance, strengths, and weaknesses of each of these modules. The detailed evaluation reports include the following:

* General description of the module and its scientific context.

* Scientific principles of operation of the algorithm.

* Program user documentation.

* Performance, strengths, and weaknesses.

* Suggestions for modifications.

* References and bibliography.

The following table summarizes the evaluation status of each of the contributed testbed software modules.

Table 2

EVALUATION STATUS OF EACH CONTRIBUTION

	CONTRIBUTION	EVALUATION OR DEMONSTRATION
CMU	PHOENIX	DETAILED EVALUATION
	Stereo/Correlation	DEMONSTRATION
MARYLAND	Relaxation	DETAILED EVALUATION
MIT	Stereo (Lisp Machine)	DEMONSTRATION
ROCHESTER	Hough Transform	DETAILED EVALUATION
STANFORD	ACRONYM	DEMONSTRATION
SRI	Road Expert	DEMONSTRATION
	RANSAC	DEMONSTRATION
	CAMDIST	DEMONSTRATION
	SHOWDTM	DEMONSTRATION
USC	Linear Features	DEMONSTRATION

C.   Summary of Evaluation Results

The following modules were evaluated in detail:

*   GHOUGH generalized Hough transform shape-finding system.
*   PHOENIX segmentation system.
*   RELAX pixel-level relaxation system.

In the evaluation process, we attempted to uncover characteristics of each system that normally become obvious to the user only through extensive experimentation. Summaries of the reports are given below:

16

1.   GHOUGH

GHOUGH uses the generalized Hough transform method to find instances of a predefined template shape in an image. It allows the location, scale, and angular rotation of the target object to be determined. The system has also been extended to detect multiple instances of the same shape in a single image.

The following templates were used in testing the program: a lake, a right angle, a circle, and an ellipse. Several interesting artifacts of the template parameterization were observed. An example was the quantization of template angles resulting from the use of discrete lattice points to compute the orientation of line segments in the template. Very dense templates generated excessive noise compared to sparser outlines because neighboring pixels were related only by angles that were multiples of 45 degrees. This significantly increases the observed noise in the estimated object parameters. Several variations of the implementation strategy have been noted that would reduce such effects.

Other significant characteristics of the algorithm were observed during attempts to locate multiple instances of circular or elliptical storage tanks in a variety of aerial imagery. A powerful feature of the Hough method is its ability to discern incomplete and occluded shapes. On the other hand, no single choice of parameters would serve to locate accurately each and every one of the circular tanks that are obvious to the human observer; the blurred nature of some of the photometry and other characteristics of the tanks (e.g., rounded tops and shadows) required that special choices of parameters and templates be made for detection of any individual tank. Thus GHOUGH was found to be very useful in detecting unique, photometrically distinguished or partial shapes, but needed higher-level information to make effective parameter choices when the available imagery was less distinctive.

17

2. PHOENIX

PHOENIX is an Ohlander-style segmentation package that uses histogram analysis to carry out a hierarchical segmentation of color imagery. Several options are available to control the number and type of the segmentation cuts performed on each histogram as well as to select criteria for determining the significance of the segmentation.

The user interface for PHOENIX is based on the CMU CI command driver, which allows a wide range of subroutines to be called in an interactive and user-controlled manner. Information about each segment of a processed image can be printed and/or displayed on the graphics system as desired. Switches and flags are available to control graphics and other output from the program. A particularly useful feature is a checkpoint system that can save the current state of a segmentation process and read it back in at a later date for more detailed examination or additional processing.

A number of fundamental properties of the PHOENIX system have been noted. The best performance is obtained for color imagery in which objects of interest have distinct colors and for which the histograms of one or more spectral bands have at least two distinct peaks. Significant region identification in deeper levels of the hierarchical process also relies on the existence of more than one distinct peak in the histograms of the parent regions. Textured monochrome images often lack these characteristics.

PHOENIX can be utilized to advantage on imagery to carry out color-based region identification if the image digitization has a rich histogram structure. Transformations of the color space may have significant effects in adapting PHOENIX to specific segmentation problems. Given appropriate original or transformed imagery, one can use the output of PHOENIX for higher-level tasks that require image segmentation information.

3.   RELAX

RELAX is a package that supports both the Hummel–Zucker–Rosenfeld and the Peleg pixel-level relaxation algorithms. To use the relaxation technique, one first assigns an initial set of probability values to the image pixels. For image-enhancement applications, there is a utility that converts a photometric image into a matrix of probabilities. A set of compatibility coefficients is then computed to support the relaxation computations. Finally, a number of relaxation iterations are performed to yield a new set of probability assignments. For image-based problems, the inverse of the original conversion utility can be run to generate a displayable grey-scale image representing the computed probability values.

The various steps in the application of the RELAX package have been integrated into a flexible user system that is based on the CI command interpreter system. We note that, for demonstration purposes, two-category relaxations allow fairly straightforward conversion between the imagery and probability structures. The usefulness of relaxation is highly dependent on the mapping from the initial imagery to the probability domain. Thus it is difficult to evaluate relaxation methods meaningfully in an abstract sense.

The RELAX system's ability to improve noisy imagery and to facilitate the extraction of image information depends strongly on the nature of the initial data and the probability assignments. The most effective way to use this system would be first to identify a subarea containing only one object of interest against a bland background, then to run RELAX to improve the signal. Alternatively, one could use an application-dependent preprocessor to assign probabilities based on criteria more complex than the values of individual pixels. This system produces excellent results if sufficient information is available for a meaningful assignment of category probabilities to the pixels of the original image, but may result in undue amplification of noise areas if the probabilities and compatibility coefficients are not chosen judiciously.

19

III   TRANSPORTABLE FEATURES OF THE TESTBED ENVIRONMENT

One of the objectives of the testbed program has been to lay the
foundation for a system that could be transported to other similar
research environments. This transportability would allow other sites to
make use of existing testbed code without having to develop their own
versions; it would also make it possible for other sites to carry out
their own evaluations and improvements of basic testbed contributions to
meet their specific needs.

These objectives have been largely met.   Each contribution to the
testbed system can be tested and demonstrated with minimal modifications
on UNIX or EUNICE/VMS VAX systems with Grinnell display devices. Many
utilities have been acquired from contributing sites or developed
locally by testbed personnel.   A new and general testbed image file
format has been created that supports all of the image types we have
found useful in integrating contributed software. A modified version of
the CMU image access package supports all essential image retrieval and
access functions.

There are also several desirable objectives that remain to be
achieved at this time. For example, graphics and image display on the
testbed are supported entirely by an extension of the CMU Grinnell
display package.   This is a large body of software whose existence
allowed basic testbed demonstration and testing objectives to be met in
a timely fashion.   However, the package is manifestly device-dependent,
so each application program carries with it the device dependence
inherent in using the Grinnell display package.   It would be desirable
to adopt a uniform device-independent graphics standard to support the
testbed demonstrations on whatever devices happen to be available at a
particular site.

Another objective is the establishment of a standard set of
utilities for registering multiple images to a ground truth data base.
Some progress has been made in this direction by the SRI RANSAC system,

the CAMDIST camera calibration system, and by the CMU "Browse" system (which is not yet ready for transport to the testbed). Further systematization of such image generation data as time of day, lighting characteristics, photometric parameters, and camera characteristics would also be desirable. The systematic application of IU techniques to cartographic tasks can only achieve its full potential when such information is available for all imagery used as source data.

In the following subsections, we present a summary of the basic capabilities that are supported in the testbed system and are potentially transportable to copies of the testbed.

A.   Utility Programs

Among the generally useful utility programs available on the testbed are the following:

(1)   CI. This is a command interpreter contributed by CMU. It allows a variety of subroutines to be linked into a top-level command processor and invoked with arguments provided interactively by the user. Extensive help and utility facilities are supplied.

(2)   ICP. This is a command interpreter for the C language contributed by SRI. It is very similar to CI, except that its treatment of arguments and local variables is more general. ICP, for example, is able to invoke system or user subroutines directly, while CI must have an argument-parsing interface written for each routine.

(3)   DOC. This is a CMU utility for generating program documentation (UNIX "man" entries) without having to know details of the TROFF phototypesetting system. All information the program needs to generate a syntactically correct "man" entry can be supplied interactively.

(4)   CONVERT. This program supports color transformations, e.g., from red-green-blue to Y-I-Q or hue-intensity-saturation spaces.

(5)   INVERT. Inverts a matrix of picture data to put the top row at the bottom, etc. This program can be used as a template for writing more general geometric or photometric transformations.

(6)   NORMALIZE. This CMU routine normalizes a grey-scale image to produce a new output image with desired compression or clipping. SRI modifications allow grey-scale stretching as well.

21

(7) REDUCE. This CMU routine extracts a subwindow of an image or rescales an image by an integer sampling factor.

(8) SHAPEUP. The original CMU routine bearing this name has been entirely rewritten to support conversions among many image formats.

(9) VIEW. This utility is a data file listing program, analogous to the UNIX "od" octal dump program. It displays files that contain integer or floating-point two-dimensional data arrays, and is particularly useful for viewing compatibility and probability files produced by the RELAX system.

B. User Interface Systems

The following systems permit useful information or features to be made available to users of the testbed:

(1) TESTBED DEMONSTRATION DIRECTORIES. Complete demonstration facilities have been set up in the testbed demonstration directory, /iu/testbed/demo. Each of the contributions is represented by a series of subdirectories supporting various informative demonstrations of program capabilities. Ground truth data for comparison with program output is also available in some cases. The command files supplied in the demonstration directories provide detailed examples of program invocation; from these examples a sophisticated user can deduce the fundamental operating procedures for each program. Detailed written documentation of program usage is available in the evaluation reports for selected contributions.

(2) VAX EMACS INFO. An INFO macro package has been developed at the SRI testbed to support an extended version of the TECO EMACS INFO system. This system is a chain-linked documentation reading and generation system that utilizes the basic window-oriented features of the EMACS editor to access, search, and display text information. On-line testbed documentation is available through the INFO system. This provides a well-structured and convenient mechanism for access to the on-line documentation of the system's functions and capabilities.

(3) LEDIT and LTAGS. Intercommunicating modified versions of of EMACS and FRANZ LISP have been implemented on the SRI EUNICE/VMS system to support Lisp-Machine-like capabilities for developing FRANZ LISP programs. LEDIT allows the user to copy any defined function from a FRANZ LISP image into an EMACS editor buffer, modify it, and then reload it into the FRANZ LISP process without

22

changing any other part of the FRANZ LISP environment. Files with many functions can be edited in EMACS and the functions of interest marked for loading when the user returns to FRANZ LISP. The LTAGS package works in concert with LEDIT in EMACS, allowing the user to display any desired function in his window for editing by simply giving the first few characters of the function name; the system automatically keeps track of which files contain which functions. The system service capabilities needed to support the intercommunications involved in LEDIT are not now available on UNIX; they therefore require the VMS operating system.

(4) ARGLIB. This is a set of utility routines for parsing program parameters and interrogating the user for additional values.

(5) PRINTERR error package. This is a testbed package that supports flexible and user-friendly reporting and handling of error conditions.

C. Picture Data Base System

The testbed Picture Data Base System (PICDBMS) is a FRANZLISP-based system that interacts with a directory of test imagery to allow the entry and retrieval of image characteristics from an image data file. Following the CMU picture file conventions, each image is assigned a named directory (e.g., /iu/tb/pic/chair) that contains the picture data (e.g., 4red.img, 4blue.img, 4green.img) along with collateral data files. PICDBMS contains utilities for creating or editing a "pic.dat" file in each picture directory. This data file, containing data formatted for easy LISP readability, includes picture descriptions, picture characteristics, and a list of data base keys. Typical data base keys that are currently supported include the labels listed in parentheses below:

* IMAGE TYPE AND MULTIPLICITY: (bw color stereo multiple)
* SCENE DOMAIN TYPE: (indoor cultural natural)
* CONTENT CHARACTERISTICS: (point linear area)
* VIEWPOINT: (aerial ground).

Other types of data can be supported as the need arises. Sets of images can be retrieved by asking for images corresponding to a set of keys;

both AND and OR conditions are supported in the data base key interrogation.

Additional facilities of PICDBMS include a browsing utility to display lists of images provided by the keyed data base retrieval subsystem. Images too large to fit on one Grinnell screen can have any desired subwindows displayed in sequence.

IV   PLANS

The future of the Image Understanding Testbed program at SRI will be closely tied to the SRI IU research efforts, as well as to the evolving characteristics of testbed copy systems to be installed at ETL and potentially at other DMA sites. The general applicability and transportability of the IU programs and utilities will continue to be enhanced as a by-product of the emerging needs of our research efforts. We anticipate that the recent incorporation of Lisp Machines into the environment will result in a substantial movement toward LISP-based IU application programs.

The major shift in emphasis in the testbed environment at SRI will be from low-level image-processing code towards increasing reliance on rule-based expert systems to guide the selection of low-level processes, the parameters to be used, and the interactive interfaces between the computer system and the human analyst. We foresee development of a substantial capability for supporting expert systems that will make it easier to apply IU research results to the solution of cartographic problems.